



TUGAS AKHIR - TF145565

**RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL PADA *HYDROSTATIC TRAINING
SYSTEM PLANT* SKALA LABORATORIUM
MENGUNAKAN AKTUATOR *MOTOR
OPERATED VALVE*.**

YUDA PAMBUDI

NRP 10511500000026

Dosen Pembimbing

Ir. Heri Joestiono, M.T

NIP. 19531116 198003 1 001



TUGAS AKHIR - TF145565

**RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL PADA *HYDROSTATIC TRAINING
SYSTEM PLANT* SKALA LABORATORIUM
MENGUNAKAN AKTUATOR *MOTOR
OPERATED VALVE*.**

**YUDA PAMBUDI
NRP 10511500000026**

**Dosen Pembimbing
Ir. Heri Joestiono, M.T
NIP. 19531116 198003 1 001**

**Ahmad Fauzan Adziimaa, S.T, M.Sc
NPP. 1991201711052**

**Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2018**



FINAL PROJECT - TF145565

**DESIGN OF LEVEL CONTROL SYSTEM IN
HYDROSTATIC TRAINING SYSTEM PLANT
LABORATORY SCALE USING MOTOR
OPERATED VALVE ACTUATOR**

YUDA PAMBUDI
NRP 10511500000026

Supervisor

Ir. Heri Joestiono, M.T
NIP. 19531116 198003 1 001

Ahmad Fauzan Adziimaa, S.T, M.Sc
NPP. 1991201711052

*Study Program DIII Instrumentation Technology
Departement of Instrumentation Engineering
Faculty of Vocation
Institut Teknologi Sepuluh Nopember
Surabaya
2018*

LEMBAR PENGESAHAN I

“RANCANG BANGUN SISTEM PENGENDALIAN *LEVEL* PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM MENGGUNAKAN AKTUATOR *MOTOR OPERATED VALVE*”

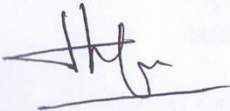
TUGAS AKHIR

Oleh :

**YUDA PAMBUDI
NRP.10511500000026**

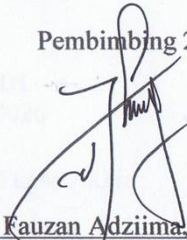
**Surabaya, 19 Juli 2018
Mengetahui / Menyetujui**

Pembimbing 1



Ir. Heri Joestiono, M.T.
NIP. 19531116 198003 1 001

Pembimbing 2



Ahmad Fauzan Adziima, S.T., M.Sc.
NPP. 1991201711052

Kepala Departemen Teknik
Instrumentasi ITS



Dr. Ir. Purwadi Agus Darwito, M.Sc
NIP. 19620822 198803 1 001

LEMBAR PENGESAHAN II

“RANCANG BANGUN SISTEM PENGENDALIAN *LEVEL* PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM DENGAN AKTUATOR *MOTOR* *OPERATED VALVE*”


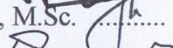
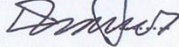
TUGAS AKHIR

Diajukan Untuk Memperoleh salah satu syarat
Untuk memperoleh gelar Ahli Madya
Pada
Program studi D3 Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

**YUDA PAMBUDI
NRP.10511500000026**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Heri Joestiono, M.T.  (Pembimbing 1)
2. Ahmad Fauzan Adziimaa, S.T., M.Sc.  (Pembimbing 2)
3. Ir. Tutug Dhanardono, M.T.  (Penguji)

**SURABAYA
2018**

**RANCANG BANGUN SISTEM PENGENDALIAN *LEVEL*
PADA HYDROSTATIC TRAINING SYSTEM *PLANT*
SKALA LABORATORIUM MENGGUNAKAN
AKTUATOR *MOTOR OPERATED VALVE***

Nama Mahasiswa : YUDA PAMBUDI
NRP : 10511500000026
Departemen : Teknik Instrumentasi FV-ITS
Dosen Pembimbing : 1. Ir. Heri Justiono, M.T
2. Ahmad Fauzan Adziima, S.T, M.Sc

ABSTRAK

Prinsip Bernoulli adalah sebuah istilah di dalam mekanika fluida yang menyatakan bahwa pada suatu aliran fluida, peningkatan pada kecepatan fluida akan menimbulkan penurunan tekanan pada aliran tersebut. Prinsip ini sebenarnya merupakan penyederhanaan dari Persamaan Bernoulli yang menyatakan bahwa jumlah energi pada suatu titik di dalam suatu aliran tertutup sama besarnya dengan jumlah energi di titik lain pada jalur aliran yang sama.. Pada perancangan sistem pengendalian level ini, sensor yang digunakan untuk mengukur *level* adalah sensor *Ultrasonic HC-SR04*. *Output* dari sensor *ultrasonic* akan mengukur *level* yang ada dan akan memberikan respon ke ATmega1284P . Saat *setpoint level* tercapai sehingga MOV sebagai aktuator akan merespon. Dari data respon pengendalian yang didapat menunjukkan bahwa pengendalian telah berjalan sesuai dengan *setpoint* yang diinginkan. Selain itu berdasarkan pengujian alat pada sistem pengendalian *level* didapatkan *respon time* yang cukup baik untuk mencapai *steady state*. Waktu yang dibutuhkan untuk mencapai *set point (settling time)* dari awal pengambilan data adalah ± 29 detik.

Kata kunci : Bernoulli, hidrostatik, pengendalian *level*.

**DESIGN OF LEVEL CONTROL SYSTEM IN
HYDROSTATIC TRAINING SYSTEM PLANT
LABORATORY SCALE USING MOTOR OPERATED VALVE
ACTUATOR**

Name of Student : YUDA PAMBUDI
NRP : 10511500000026
Department : Teknik Instrumentasi FV-ITS
Supervisor : 1. Ir. Heri Justiono, M.T
2. Ahmad Fauzan Adziima, S.T, M.Sc

ABSTRACT

The Bernoulli principle is a term in fluid mechanics which states that in a fluid flow, an increase in fluid velocity will cause a decrease in pressure on the flow. This principle is actually a simplification of the Bernoulli Equation which states that the amount of energy at a point in a closed flow is equal to the amount of energy at another point on the same flow path. In the design of this level control system, the sensor used to measure level is Ultrasonic HC-SR04 sensor. The output of the ultrasonic sensor will measure the current level and will respond to the ATmega1284P. When level setpoint is reached so that MOV as actuator will respond. From the control response data obtained shows that the control has been running in accordance with the desired setpoint. In addition, based on the testing of tools in the system of level control, the response time is good enough to achieve steady state. The time required to reach the set point (settling time) from the initial data retrieval is ± 29 seconds.

Keywords: Bernoulli, hydrostatic, level control

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan YME atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “RANCANG BANGUN SISTEM PENGENDALIAN *LEVEL* PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM MENGGUNAKAN AKTUATOR *MOTOR OPERATED VALVE*”. Penulis telah banyak mendapatkan bantuan dari berbagai pihak dalam menyelesaikan Tugas Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Dr.Ir.Purwadi Agus D, M.Sc selaku Ketua Departemen D3 Teknik Instrumentasi, dan Bapak Dr. Bambang Lelono Widjiantoro, ST, MT, selaku dosen wali penulis yang telah sabar memberikan dukungan, bimbingan, serta ilmu yang sangat bermanfaat.
2. Bapak Ir. Heri Justiono, M.T selaku dosen pembimbing I dan Bapak Ahmad Fauzan Adziimaa, S.T, M.Sc yang senantiasa memberikan motivasi, bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini.
3. Bapak dan Ibu dosen Departemen Teknik Instrumentasi yang telah memberikan ilmu selama kuliah.
4. Seluruh Staf Departemen Teknik Instrumentasi yang telah membantu penulis dalam hal administrasi.
5. Ibu tercinta ibu G.K. Irianti, Bapak Dwi Mulyono, Mas Herman, Mba Tyas, Mas Tobit dan seluruh keluarga besar tercinta yang senantiasa memberikan dukungan, semangat dan doa kepada penulis.
6. Teman-teman seperjuangan satu kelompok dalam mengerjakan Tugas Akhir Ardy, Izah, Arul, Tito, dan Mba Andiah, Serta teman-teman D3-Teknik Instrumentasi lainnya serta teman-teman angkatan F50 yang selalu memotivasi penulis.

7. Teman-teman SONIC LA16 PSM ITS yang tidak bisa penulis sebutkan satu persatu.
8. Teman-teman yang mendukung saya yang tidak dapat saya sebutkan satu persatu.

Penulis menyadari bahwa penulisan laporan Tugas Akhir ini tidaklah sempurna. Oleh karena itu sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik lagi. Penulis juga berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembacanya.

Surabaya, 18 Juli 2018

Penulis.

Penulis.

DAFTAR ISI

	Hal
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	iii
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii

BAB I. PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3

BAB II. DASAR TEORI

2.1 Fluida	5
2.2 Tekanan Hidrostatik	5
2.3 Sistem Kontrol dan Pengendalian Proses	6
2.4 Pompa Air	8
2.5 Sensor Ultrasonik	8
2.6 Sensor Ultrasonik HC-SR04	9
2.7 Orifice	11
2.8 Mikrokontroler ATmega1284P.....	12
2.9 Motor Servo Controlled Valve	15

BAB III. PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir (<i>Flowchart</i>)	19
3.2 Studi Literatur	19
3.3 Perancangan Sistem	19

3.4 Pembuatan Alat	21
3.5 Pengujian Alat dan Analisis	24
3.6 Pembuatan Laporan	25

BAB IV. HASIL DAN PEMBAHASAN

4.1 Hasil Percobaan	27
4.2 Pembahasan	41

BAB V. PENUTUP

5.1 Kesimpulan.....	43
5.2 Saran.....	43

DAFTAR PUSTAKA

LAMPIRAN A (*Listing Program Pada Code Vision Avr*)

LAMPIRAN B (*Data Sheet System Atmega1284P*)

LAMPIRAN C (*Data Sheet Sensor Ultrasonic HC-SR04*)

LAMPIRAN D (*Data Sheet Servo MKS HV777*)

BIODATA PENULIS

DAFTAR GAMBAR

	Hal
Gambar 2.1 Sensor Ultrasonik HC-SR04	9
Gambar 2.2 Waktu tempuh gelombang ultrasonik	10
Gambar 2.3 Timing Diagram sensor ultrasonik.....	11
Gambar 2.4 <i>Orifice Plate</i>	11
Gambar 2.5 Skematik <i>Orifice Plate</i>	12
Gambar 2.6 Konfigurasi Pin ATMega1284p-PU.....	14
Gambar 2.7 Konfigurasi Pin ATMega1284p-AU.....	14
Gambar 2.8 Motor Servo	15
Gambar 2.9 Komponen Motor Servo	16
Gambar 2.10 Pulsa pada Motor Servo	16
Gambar 3.1 <i>Flowchart</i> Tugas Akhir	19
Gambar 3.2 Blog Diagram Pengendalian	20
Gambar 3.3 <i>Flowchat</i> Algoritma.....	21
Gambar 3.4 PFD Plant	22
Gambar 3.5 Penjelasan symbol PFD	22
Gambar 3.6 P&ID Sistem Pengendalian	23
Gambar 3.7 Penjelasan symbol P&ID.....	23
Gambar 4.1 Hasil Rancang Bangun Plant	27
Gambar 4.2 Pengendalian Level-Flow	27
Gambar 4.3 Grafik Perbandingan sensor Ultrasonik.....	34
Gambar 4.4 Grafik respon sistem saat setpoint 26.....	38
Gambar 4.5 Grafik respon sistem saat setpoint 25.....	40

DAFTAR TABEL

	Hal
Tabel 4.1 Hasil Pengujian Naik Sensor Ultrasonik	28
Tabel 4.2 Hasil Pengujian Turun Sensor Ultrasonik	30
Tabel 4.3 Rata-rata pembacaan sensor Ultrasonik	32
Tabel 4.4 Hasil Uji Motor Servo Valve	36
Tabel 4.5 Hasil Uji Set Point 26.....	37
Tabel 4.6 Hasil Uji Set Point 26.....	39

BAB I

PENDAHULUAN

1.1 Latar Belakang

Prinsip Bernoulli adalah sebuah istilah di dalam mekanika fluida yang menyatakan bahwa pada suatu aliran fluida, peningkatan pada kecepatan *fluida* akan menimbulkan penurunan tekanan pada aliran tersebut. Prinsip ini sebenarnya merupakan penyederhanaan dari Persamaan Bernoulli yang menyatakan bahwa jumlah energi pada suatu titik di dalam suatu aliran tertutup sama besarnya dengan jumlah energi di titik lain pada jalur aliran yang sama. Prinsip ini diambil dari nama ilmuwan Belanda bernama Daniel Bernoulli. Prinsip ini banyak diterapkan pada kehidupan sehari-hari maupun di industri sehingga diperlukan pemahaman dalam percobaan praktikum yang menjadi latar belakang dibuatnya tugas akhir Rancang Bangun *Hydrostatic Training System Plant* ini. (Adam Thai, 2016)

Dalam Rancang Bangun *Hydrostatic Training System Plant* ini memiliki sistem pengendalian *level* menggunakan aktuator MOV. Ketinggian dari *fluida* dalam tangki perlu dilakukan pengamatan atau pengukuran agar *fluida* dalam tangki tetap dalam batas atau *set point* yang diinginkan, selanjutnya apabila ketinggian dari *fluida* sudah mencapai *set point* yang diinginkan maka akan dilakukan pengendalian *level* dengan mengatur bukaan dari *motor valve* agar tetap pada *set point* yang diinginkan.

Dalam Rancang Bangun *Hydrostatic Training System Plant* skala laboratorium ini menggunakan sistem tangki tertutup dimana tidak bisa dilakukan pengamatan atau pengukuran secara langsung. Dalam melakukan pengukuran *level* untuk mengetahui tinggi air pada suatu tempat tertutup diperlukan adanya pengendalian menggunakan sensor ultrasonik. Sensor ultrasonik memiliki kelebihan yaitu dalam pengukuran tidak bersentuhan langsung dengan *fluida* yang diukur. Selain itu untuk mengatur *fluida* yang masuk maka diperlukan aktuator *motor operated valve* dimana nanti bukaan dari *valve* tersebut akan diatur menggunakan mikrokontroler ATmega1284P. Oleh karena itu, dibuat tugas akhir dengan judul “RANCANG BANGUN SISTEM

PENGENDALIAN *LEVEL* PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM MENGGUNAKAN AKTUATOR *MOTOR OPERATED VALVE*”

1.2 Perumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah:

- a. Bagaimana cara merancang sistem pengendalian *level* pada *Hydrostatic Training System Plant* skala laboratorium?
- b. Bagaimana merealisasikan sistem pengendalian *level* menggunakan mikrokontroller Atmega1284P dan sensor ultrasonik HC-SR04 dan dalam sistem kontrol *level* pada *Hydrostatic Training System Plant* skala laboratorium ?

1.3 Batasan Masalah

Adapun batas ruang lingkup dari penelitian tugas akhir ini yaitu :

1. Pengendalian *level* pada *Hydrostatic Training System Plant* pada tangki 1 skala laboratorium berbasis Mikrokontroller Atmega1284P
2. Sistem pengendalian *level* menggunakan aktuator *motor operated valve* dan sensor ultrasonik HC-SR04.

1.4 Tujuan

Tujuan utama dari rancang bangun alat ini adalah untuk memenuhi mata kuliah tugas akhir sebagai syarat kelulusan dari Program Studi DIII Teknologi Instrumentasi serta hasil dari penelitian tersebut diharapkan dapat digunakan sebagai alat penunjang praktikum dan untuk memberikan solusi pada rumusan masalah diatas adalah sebagai berikut :

- a. Membuat rancang sistem pengendalian *level* pada *Hydrostatic Training System Plant* skala laboratorium menggunakan mikrokontroller Atmega1284P dan sensor ultrasonik HC-SR04 dalam sistem kontrol *level* dan *motor operated valve* sebagai aktuator.

- b. Merealisasikan sistem pengendalian *level* menggunakan aktuator *motor operated valve* dan sensor ultrasonik HC-SR04 dan dalam sistem kontrol *level* pada *Hydrostatic Training System Plant* skala laboratorium.

1.5 Manfaat

Manfaat yang didapat dari tugas akhir ini adalah jika plant *Hydrostatic Training System* dapat terealisasi dapat digunakan untuk metode pembelajaran dan praktikum matakuliah Sistem Pengukuran Aliran, Sistem Instrumentasi Industri dan beberapa mata kuliah yang ada di Departemen Teknik Instrumentasi.

(Halaman ini sengaja dikosongkan)

BAB II

DASAR TEORI

2.1 Fluida

Fluida dibedakan menjadi dua, yaitu fluida cair dan gas. Partikel-partikel fluida yang bergerak disebut aliran fluida. Untuk mengetahui besar aliran fluida digunakan hukum kontinuitas Bernoulli (M. A. Crabtree, 2009)

Terdapat tiga jenis aliran fluida dalam pipa, yaitu aliran laminar, transisional dan turbulenta yang dapat diketahui menggunakan bilangan Reynold (Re).

Berikut ini adalah jenis-jenis aliran fluida:

1) Aliran Laminar

Merupakan partikel-partikel fluida bergerak lancar dengan kecepatan rendah dan viskositas tinggi.

2) Aliran Transisional

Merupakan partikel-partikel fluida bergerak lancar kemudian berubah acak tidak stabil. Hal tersebut dikarenakan adanya perubahan kecepatan rendah ke tinggi dan perubahan viskositas tinggi ke rendah.

3) Aliran Turbulent

Merupakan partikel-partikel fluida bergerak acak tidak stabil dengan kecepatan tinggi dan viskositas rendah.

2.2 Tekanan Hidrostatik

Tekanan Hidrostatik adalah tekanan yang diakibatkan oleh gaya yang ada pada zat cair terhadap suatu luas bidang tekan pada kedalaman tertentu. Besarnya tekanan ini bergantung kepada ketinggian zat cair, massa jenis dan percepatan gravitasi. Tekanan Hidrostatik hanya berlaku pada zat cair yang tidak bergerak. Sedangkan tekanan zat cair yang bergerak akan dipelajari lebih lanjut dalam Mekanika Fluida. Tekanan pada zat cair secara umum dibedakan menjadi dua jenis tekanan, yakni tekanan zat cair yang tidak bergerak (tekanan hidrostatik) dan tekanan zat cair yang bergerak (mengalir). (Ibadurrahman, 2017)

Tekanan Hidrostatik merupakan tekanan yang dipengaruhi oleh *level* fluida. Arah gaya tekan yang ke semua arah menghasilkan persamaan berikut.:(R. H. Perry, D. W. Green, 1997)

$$P = \rho \times g \times h \dots\dots\dots (2.1)$$

Keterangan:

P= Tekanan Hidrostatik (N/m²)

ρ = Massa Jenis (kg/m³)

g= Percepatan gravitasi (m/det²)

h= Kedalaman/ketinggian (m)

2.3 Sistem Kontrol dan Pengendalian Proses

Hampir semua proses di industry membutuhkan peralatan-peralatan otomatis untuk mengendalikan parameter –parameter prosesnya. Otomatisasi tidak hanya diperlukan demi kelancaran operasi, keamanan, ekonomi maupun mutu produk, tetapi lebih merupakan kebutuhan pokok. Suatu proses industri tidak akan dapat dijalankan tanpa bantuan system pengendalian. Parameter - parameter yang umum dikendalikan dalam suatu proses adalah tekanan (pressure), laju aliran (flow), suhu (temperature) dan permukaan zat cair (level). Gabungan serta alat-alat pengendalian otomatis itulah yang disebut system pengendalian proses (process control system). Sedangkan semua peralatan yang membentuk sistem pengendalian tersebut disebut instrumentasi pengendalian proses (process control instrumentation).Istilah – istilah yang perlu diketahui dalam sistem otomatis adalah sebagai berikut:

- Proses : tatanan peralatan yang mempunyai suatu fungsi tertentu, contohnya level tangki.
- Controlled Variable : besaran atau variabel yang dikendalikan. Besaran ini pada diagram kotak disebut juga dengan output proses atau level tangki..
- Manipulated Variable : input dari suatu proses yang dapat di manipulasi agar controlled variable sesuai dengan set point-nya.

- Disturbance : besaran lain (selain manipulated variable) yang dapat menyebabkan berubahnya level tangki diatas dari tangki yang dikendalikan
- Sensing Element : bagian paling ujung suatu sistem pengukuran, seperti sensor level.
- Transmitter : untuk membaca sinyal sensing element dan mengubah sinyal yang dapat dipahami oleh controller seperti signal conditioning (SC).
- Transducer : unit pengalih sinyal.
- Measurement Variable : sinyal yang keluar dari transmitter. Jika dalam pengendalian level, sinyal yang keluar adalah berupa level.
- Set Point : besar process variable (level) yang dikehendaki.
- Controller : elemen yang melakukan tahapan mengukur - membandingkan - menghitung - mengkoreksi. Proporsional Integral Derivatif (PID) salah satu controller yang sering digunakan dalam industri.
- Final Control Element : bagian akhir dari instrumentasi sistem pengendalian yang berfungsi untuk mengubah measurement variable dengan cara manipulated variable, berdasarkan perintah pengendali. Salah satu final control element yang digunakan dalam pengendalian level adalah motorized valve.

Secara garis besar suatu rangkaian pengendalian proses dibagi menjadi 4 langkah, yaitu : mengukur-membandingkan-menghitung-mengkoreksi. Langkah pertama yaitu mengukur , merupakan tugas dari sensor. Langkah berikutnya adalah membandingkan apakah hasil pengukuran dari sensor sudah sesuai dengan apa yang dikehendaki. Apabila terjadi ketidaksesuaian antara set point dengan hasil pengukuran maka akan dilakukan perhitungan untuk menentukan aksi apa yang dilakukan supaya sesuai dengan set point yang diinginkan. Pada langkah kedua dan ketiga ini adalah tugas dari pengendali. Langkah terakhir adalah melakukan pengkoreksian yang merupakan tugas dari aktuator. (Danel, 2012)

2.4 Pompa Air

Pompa adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (suction) dengan bagian keluar (discharge). (Iwan Nugraha, 2014)

Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran. Akibat tekanan yang rendah pada sisi isap pompa maka fluida akan naik dari kedalaman tertentu, sedangkan akibat tekanan yang tinggi pada sisi discharge akan memaksa fluida untuk naik sampai pada ketinggian yang diinginkan.

2.5 Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang memanfaatkan pancaran gelombang ultrasonik. Sensor ultrasonik ini terdiri dari rangkaian pemancar ultrasonik yang disebut transmitter dan rangkaian penerima ultrasonik disebut receiver. Gelombang ultrasonik merupakan gelombang akustik yang memiliki frekuensi mulai 20 kHz hingga sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fasa gas, cair hingga padat. Jika gelombang ultrasonik berjalan melalui sebuah medium, Secara matematis besarnya jarak dapat dihitung sebagai berikut:

$$s = v.t/2 \dots\dots\dots (2.2)$$

Dimana s adalah jarak dalam satuan meter, v adalah kecepatan gelombang suara yaitu 344 m/detik dan t adalah waktu tempuh dalam satuan detik. Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan.

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut piezoelektrik. Piezoelektrik ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. Sensor ultrasonik secara umum digunakan untuk suatu pengungkapan tak sentuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang kearah sensor.

Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium. Rangkaian penyusun sensor ultrasonik ini terdiri dari transmitter, reiceiver, dan komparator.

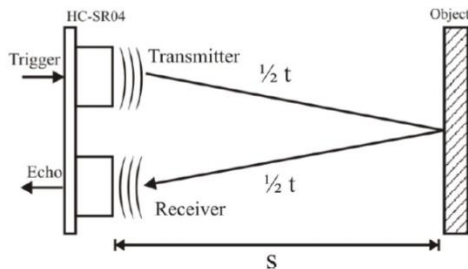
2.6 Sensor Jarak Ultrasonik HC-SR04

Sensor jarak ultrasonic HC-SR04 adalah sensor 40 KHz. HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor.



Gambar 2.1 Sensor Ultrasonik HC-SR04(Martani and Endarko, 2014)

HC-SR04 memiliki 2 komponen utama sebagai penyusunnya yaitu *ultrasonic transmitter* dan *ultrasonic receiver*. Fungsi dari *ultrasonic transmitter* adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian *ultrasonic receiver* menangkap hasil pantulan gelombang ultrasonik yang mengenai suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul. (Trisnobudi, 2001)

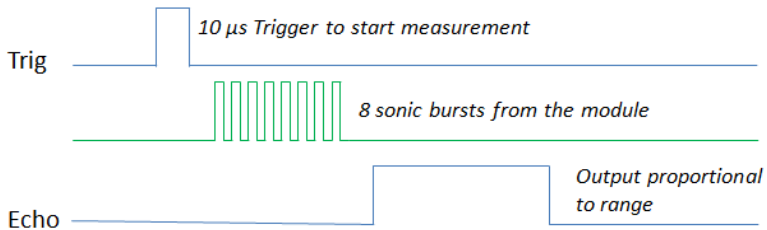


Gambar 2.2 Waktu tempuh gelombang ultrasonik

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah, ketika pulsa *trigger* diberikan pada sensor, *transmitter* akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan output TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah *receiver* menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan output TTL transisi turun.

Pemilihan HC-SR04 sebagai sensor jarak yang akan digunakan pada penelitian ini karena memiliki fitur sebagai berikut; kinerja yang stabil, pengukuran jarak yang akurat dengan ketelitian 0,3 cm, pengukuran maksimum dapat mencapai 4 meter dengan jarak minimum 2 cm, ukuran yang ringkas dan dapat beroperasi pada level tegangan TTL Prinsip pengoperasian sensor ultrasonik HC-SR04 adalah sebagai berikut ; awali dengan memberikan pulsa Low (0) ketika modul mulai dioperasikan, kemudian berikan pulsa High (1) pada trigger selama 10 μ s

sehingga modul mulai memancarkan 8 gelombang kotak dengan frekuensi 40 KHz, tunggu hingga transisi naik terjadi pada output dan mulai perhitungan waktu hingga transisi turun terjadi. (Trisnobudi, 2001)



Gambar 2.3 Timing diagram pengoperasian sensor ultrasonik HC-SR04

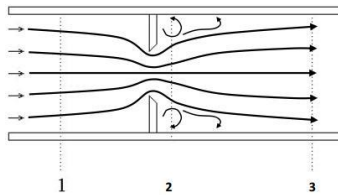
2.7 Orifice

Orifice plate adalah pelat datar dengan lubang konsentrik bersisi tajam yang didesain dengan dimensi khusus yang membentuk penghalang yang terkalibrasi terhadap aliran gas alam yang melalui meter tube sehingga menyebabkan perbedaan tekanan antara tekanan sebelum dan sesudah melewati orifice tersebut. *Orifice* merupakan salah satu komponen dari perangkat primer (*primary device*) untuk mengukur aliran dengan menggunakan prinsip mengubah kecepatan aliran, riilnya yaitu mengubah luasan yang dilalui aliran fluida tersebut (*orifice*). (Haidi, no date)



Gambar 2.4 Orifice Plate

Pada dasarnya *orifice* berupa plat tipis dengan lubang di bagian tertentu (umumnya di tengah). Fluida yang mengalir melalui pipa ketika sampai pada *orifice* akan dipaksa untuk melewati lubang pada *orifice*. Hal itu menyebabkan terjadinya perubahan kecepatan dan tekanan. Titik dimana terjadi kecepatan maksimum dan tekanan minimum disebut *vena contracta*. Setelah melewati *vena contracta* kecepatan dan tekanan akan mengalami perubahan lagi. Dengan mengetahui perbedaan tekanan pada pipa normal dan tekanan pada *vena contracta*, laju aliran volume dan laju aliran massa dapat diperoleh dengan persamaan *Bernoulli*. (Zimmermann, 1999)



Gambar 2.5 Skematik *Orifice Plate*

Keterangan :

- 1 Fluida memasuki orifice pada P1 dan V1
- 2 Area yang lebih kecil mengakibatkan kecepatan meningkat sehingga terjadi pressure drop sebesar $P1 - P2$
- 3 Pada titik 3 tekanan kembali meningkat menjadi P3, namun P3 tetap lebih kecil daripada P1. Ini disebut non recoverable pressure drop (penurunan tekanan yang tidak bisa dipulihkan)

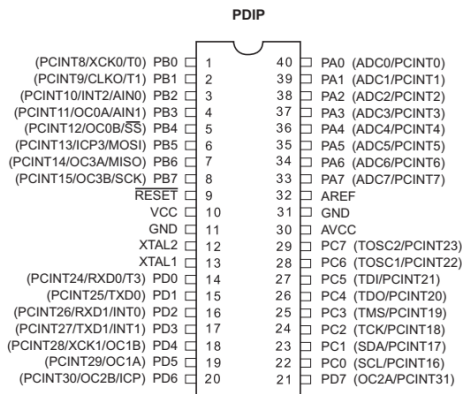
2.8 Mikrokontroler ATmega1284P

ATmega1284P adalah mikrokontroler CMOS 8-bit berdaya rendah berdasarkan pada AVR ditingkatkan arsitektur RISC. Dengan mengeksekusi instruksi yang kuat dalam satu siklus clock, ATmega1284P mencapai *throughput* mendekati 1 MIPS per MHz memungkinkan perancang sistem untuk mengoptimalkan konsumsi daya dibandingkan kecepatan pemrosesan. Mikrokontroler ini memiliki SRAM dan EEPROM yang besar

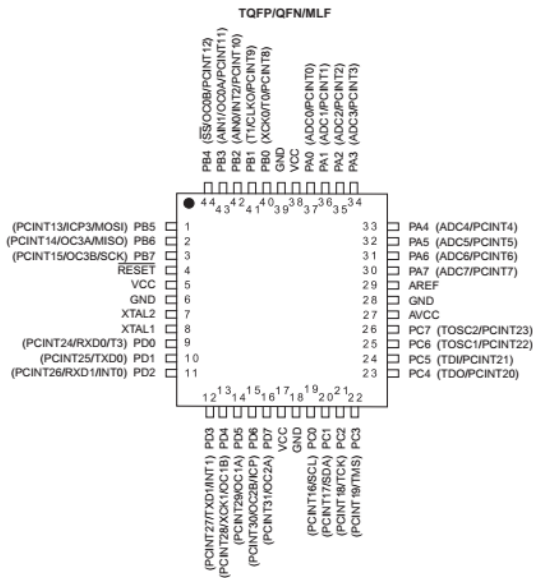
yaitu 16KB dan 4KB. Inti AVR menggabungkan set instruksi yang kaya dengan 32 register kerja tujuan umum. Semua 32 register secara langsung terhubung ke Arithmetic Logic Unit (ALU), memungkinkan dua register independen untuk diakses dalam satu instruksi tunggal yang dijalankan dalam satu siklus clock. Arsitektur yang dihasilkan lebih efisien kode saat mencapai throughput hingga sepuluh kali lebih cepat daripada mikrokontroler CISC konvensional.

ATmega1284P menyediakan fitur-fitur berikut: 128K byte In-System Programmable Flash dengan kemampuan Read-While-Write, EEPROM 4K byte, SRAM 16K byte, 32 jalur I / O tujuan umum, 32 register kerja untuk tujuan umum, Real Time Counter (RTC)), tiga Timer / Penghitung fleksibel dengan mode perbandingan dan PWM, 2 USART, 2-kawat Serial Interface berorientasi byte, ADC 8-channel, 10-bit dengan tahap input diferensial opsional dengan gain yang dapat diprogram, Pengatur Waktu Pengawas yang dapat diprogram dengan Oscillator Internal, port serial SPI, IEEE std. 1149.1 antarmuka uji JTAG yang sesuai, juga digunakan untuk mengakses sistem Debug On-chip dan pemrograman serta enam mode penghematan daya perangkat yang dapat dipilih. Mode Idle menghentikan CPU sementara memungkinkan SRAM, Timer / Penghitung, port SPI, dan sistem interupsi untuk terus berfungsi. Mode Power-down menyimpan isi register tetapi membekukan Oscillator, menonaktifkan semua fungsi chip lainnya sampai interupsi berikutnya atau Reset Perangkat Keras. Dalam mode Hemat daya, penghitung waktu asinkron terus berjalan, memungkinkan pengguna untuk mempertahankan basis waktu sementara perangkat lainnya tidur. Mode ADC Noise Reduction akan menghentikan CPU dan semua modul I / O kecuali Asynchronous Timer dan ADC, untuk meminimalkan gangguan pengalihan selama konversi ADC. Dalam mode Siaga, Crystal / Resonator Oscillator sedang berjalan saat sisa perangkat sedang tidur. Hal ini memungkinkan start-up yang sangat cepat dikombinasikan dengan konsumsi daya yang rendah. Dalam mode Standby Diperpanjang, baik *themin*

Oscillator dan Asynchronous Timer terus berjalan. (Channels, no date)



Gambar 2.6 Konfigurasi Pin ATmega1284P-PU



Gambar 2.7 Konfigurasi Pin ATmega1284P-AU

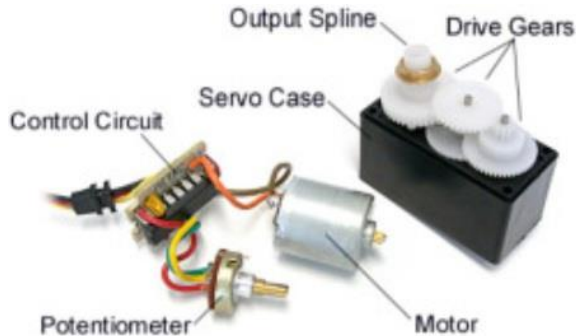
2.9 Motor Servo Controlled Valve

Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotor-nya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo.



Gambar 2.8 Motor Servo (SERVO, 2018)

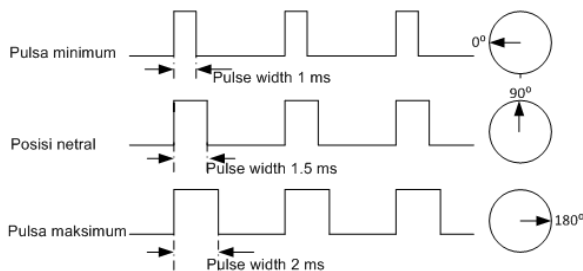
Motor servo pada dasarnya dibuat menggunakan motor DC yang dilengkapi dengan controler dan sensor posisi sehingga dapat memiliki gerakan 0° , 90° , 180° atau 360° . Berikut adalah komponen internal sebuah motor servo 180° .



Gambar 2.9 Komponen Motor Servo (Suprianto, 2015)

Tiap komponen pada motor servo diatas masing-masing memiliki fungsi sebagai controller, driver, sensor, girbox dan aktuator. Pada gambar diatas terlihat beberapa bagian komponen motor servo. Motor pada sebuah motor servo adalah motor DC yang dikendalikan oleh bagian controler, kemudian komponen yang berfungsi sebagai sensor adalah potensiometer yang terhubung pada sistem girbox pada motor servo.

Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90° , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180° . (Prakoso Galih, 2017)



Gambar 2.10 Pulsa pada Motor Servo (Abdul M, 2017)

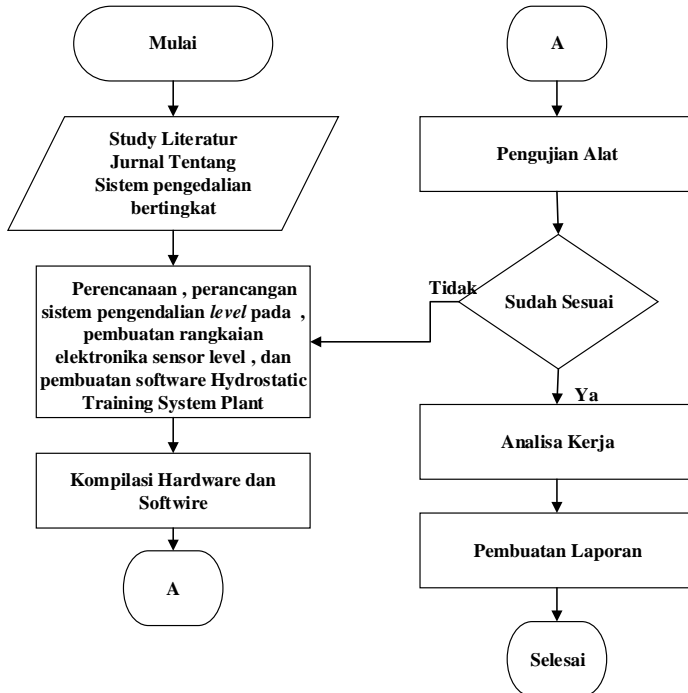
Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0° / netral).

Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton duty cycle, dan bertahan diposisi tersebut. (Howard Eglowstein, 2012)

(Halaman ini sengaja dikosongkan)

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir (Flowchart)



Gambar 3.1 *Flowchart* Tugas Akhir

3.2 *Study literatur sistem pengendalian level*

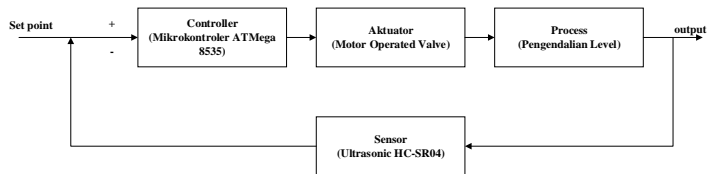
Pada tahap ini akan dilakukan pencarian dasar teori yang berkaitan dan mendukung sistem pengendalian *level* pada alat *Hydrostatic Training System Plant* skala laboratorium . Teori yang dicari ultrasonik, ATmega1284P, dan sistem pengendalian *level*.

3.3 Perancangan Sistem

Pada tahap ini di lakukan perancangan sistem pengendalian *level* pada tangki 1. Perancangan sistem akan menjelaskan tentang perancangan-perancangan eletrik seperti rangkaian pengkondisian

sinyal dan perancangan mekanik alat. Berikut ini adalah diagram blok dari Pengendalian cascade dari *hydrostatic training system plant*.

Berikut merupakan Blog Diagram dari Sistem Pengendalian *level* pada *Hydrostatic Training System Plant* :

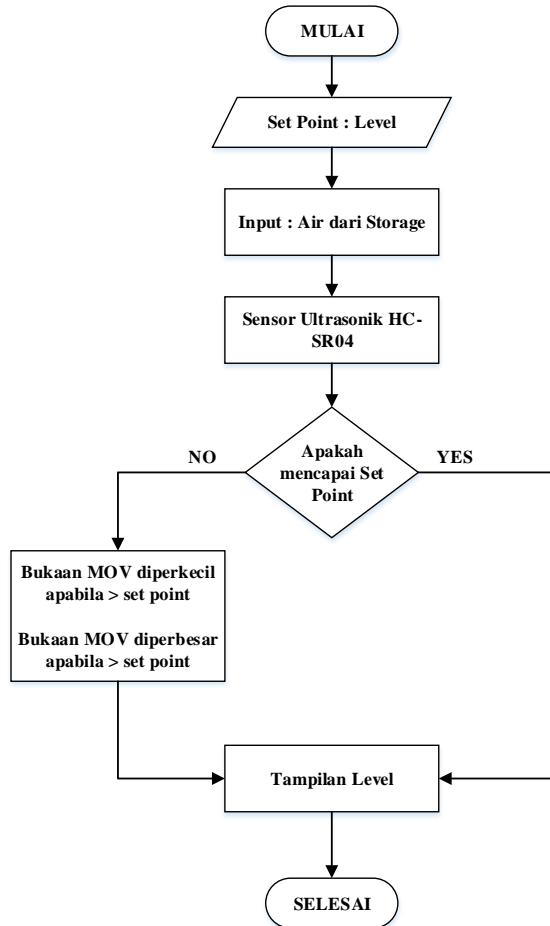


Gambar 3.2 Blok Diagram pengendalian *level*

Penjelasan tentang Blog Diagram Pengendalian *level* pada *Hydrostatic Training System Plant* adalah sebagai berikut :

1. Sensor yang digunakan yaitu sensor ultrasonic HC-SR04 yang mempunyai rentang pengukuran 2 cm – 400 cm dengan ketelitian 0,3 cm dimana digunakan untuk mengetahui *level* yang ada pada setiap tangki. *Set point* yang akan dideteksi oleh sensor ada 3, yaitu set point 1, set point 2 dimana nantinya safety akan bekerja menyalakan buzzer, dan set point 3 berfungsi untuk membuka MOV untuk membuang fluida kembali ke *storage*.
2. Kontroler yang digunakan yaitu Mikrokontroler ATmega1284P dimana berfungsi sebagai pengatur jalannya proses pengendalian *level* pada *Hydrostatic Training System Plant*. Jika sensor memberikan sinyal bahwa *level* yang ada pada tangki tidak sesuai dengan set point maka kontroler akan memberikan sinyal pada aktuator untuk melakukan aksi agar sesuai dengan set point
3. Aktuator yang digunakan yaitu *Motor Operated Valve* yang berfungsi jika diberikan sinyal oleh kontroler untuk melakukan aksi buka tutup agar fluida pada tangki sesuai dengan set point.

4. Proses yang terjadi yaitu pengendalian *level* pada setiap tangki.



Gambar 3.3 *Flowchart* Algoritma Sistem Pengendalian *Level*

3.4 Pembuatan Alat

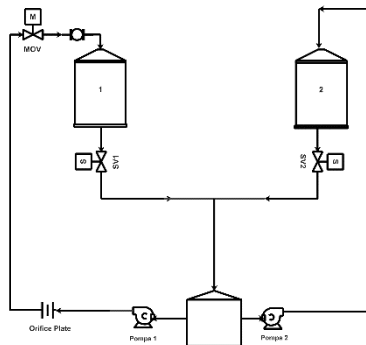
Pada tahap pembuatan alat ini akan dibagi menjadi 2 bagian, yaitu pembuatan hardware alat dan pembuatan software alat. Untuk pembuatan hardware terdiri dari pembuatan elektik dan mekanik

pada alat, sedangkan pembuatan software berisi langkah-langkah cara mengkodng mikrokontroler agar sistem control dapat berjalan sesuai dengan yang diinginkan.

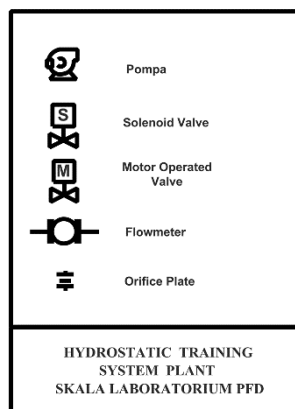
Berikut ini adalah rincian pembuatan model *Hydrostatic Training System Plant* skala laboratorium :

Process Flow Diagram (PFD)

Berikut merupakan PFD dari alat *Hydrostatic Training System Plant* skala laboratorium:



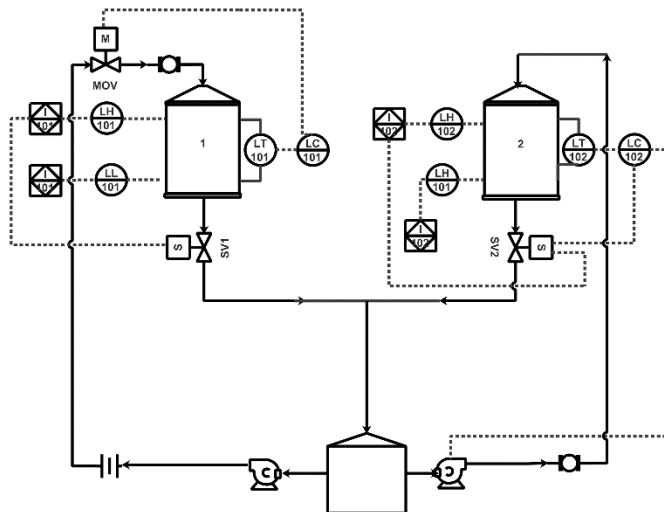
Gambar 3.4 PFD Plant



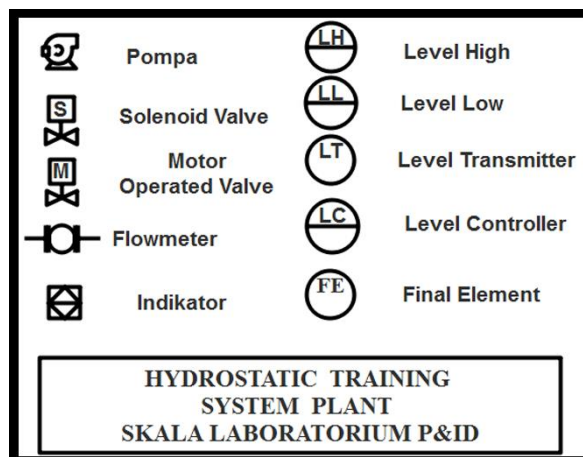
Gambar 3.5 Penjelasan Simbol Pada PFD

Piping and Instrument Diagram (P&ID)

Berikut merupakan P&ID dari Sistem Pengendalian *Leve Hydrostatic Training System Plant* skala laboratorium :



Gambar 3.6 P&ID Sistem Pengendalian *Level*



Gambar 3.7 Penjelasan Simbol P&ID

Pengendalian *level* pada *Hydrostatic Training System Plant* skala laboratorium ini menggunakan sensor Ultrasonik HC-SR04 untuk sensor *level* dimana rentang pengukurannya 2 cm sampai dengan 400 cm dengan ketelitian 0,3 cm.

Spesifikasi komponen pada *Hydrostatic Training System Plant*

No	Data alat	Spesifikasi
1	Aliran listrik	Tegangan : 230 Volt Frekuensi : 50 – 60 Hz
2	Pompa	Daya : 250W Max. Laju aliran : 9m ³ /h
3	Tangki Penyimpanan	Tinggi : 500 mm Diameter Tanki 1 : 100 mm Max tekanan : 0,0049 bar Diameter Tanki 2 : 133 mm Diameter Tanki 3 : 200 mm
4	Jangkauan pengukuran	Tekanan : 1 – 1,5 bar Tekanan diferensial : 0 – 500 mmWS Tekanan diferensial : 0 – 0,4 bar Desnitas : 1x0,8 – 1 g/cm ³

3.5 Pengujian Alat dan Analisis

Pada tahap pengujian alat ini dilakukan percobaan menggunakan kompresor sebagai laju aliran input dan dibandingkan dengan rotameter sebagai alat yang sudah standart untuk mengetahui nilai range, span, histerisis, akurasi serta kesalahan yang ada pada alat tersebut. Analisa Data dilakukan dengan cara melakukan pengambilan data dan data tersebut diolah

agar didapatkan nilai sesuai dengan karakteristik maupun rumusan yang telah ditentukan.

3.6 Pembuatan Laporan

Pada tahap ini dilakukan analisa data apakah sudah sesuai kriteria atau belum dengan membandingkan dengan perhitungan sesuai rumus yang dilakukan diawal. Jika tidak sesuai maka akan dianalisa bagian mana yang menyebabkan tidak sesuainya. Setelah dianalisa kemudian didapatkan kesimpulan dan dituliskan dalam laporan tugas akhir.

(Halaman ini sengaja dikosongkan)

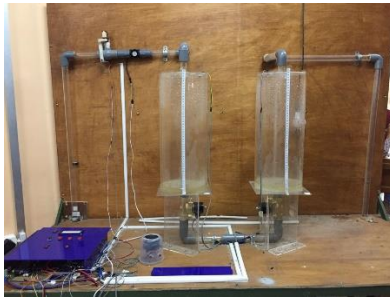
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Percobaan

4.1.1 Hasil Rancang Bangun Plant

Berikut ini adalah hasil dari perancangan Hidrostatik Training System Plant Skala Laboratorium :



Gambar 4.1 Hasil Rancang Bangun Plant

4.1.2 Pembuatan Sistem Pengendalian Level

Berikut ini adalah hasil dari perancangan Sistem pengendalian Level pada Hidrostatik Training System Plant Skala Laboratorium :



Gambar 4.2 Pengendalian Level

4.1.3 Pengujian Sensor Ultrasonik HC-SR04

Pada penelitian tugas akhir ini menggunakan sensor ultrasonik HC-SR04 yang memiliki range 2 cm – 4 m pada *datasheet*. Berikut merupakan data pengukuran dari pengujian sensor ultrasonic :

Tabel 4.1 Hasil Pengujian Naik Sensor Ultrasonik

JARAK 5 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	5.51	5	-0.51	-10.2
2	5.38	5	-0.38	-7.6
3	6.12	5	-1.12	-22.4
4	5.64	5	-0.64	-12.8
5	5.83	5	-0.83	-16.6
JARAK 10 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	10.51	10	-0.51	-5.1
2	10.32	10	-0.32	-3.2
3	10.54	10	-0.54	-5.4
4	11.05	10	-1.05	-10.5
5	10.89	10	-0.89	-8.9
JARAK 15 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	15.43	15	-0.43	-2.866666667
2	15.52	15	-0.52	-3.466666667
3	15.62	15	-0.62	-4.133333333
4	15.64	15	-0.64	-4.266666667
5	15.45	15	-0.45	-3

Tabel 4.1 Lanjutan

JARAK 20 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	20.53	20	-0.53	-2.65
2	20.56	20	-0.56	-2.8
3	21.1	20	-1.1	-5.5
4	21.05	20	-1.05	-5.25
5	20.65	20	-0.65	-3.25
JARAK 25 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	25.51	25	-0.51	-2.04
2	26.01	25	-1.01	-4.04
3	25.87	25	-0.87	-3.48
4	25.65	25	-0.65	-2.6
5	25.89	25	-0.89	-3.56
JARAK 30 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	30.66	30	-0.66	-2.2
2	31.05	30	-1.05	-3.5
3	30.56	30	-0.56	-1.866666667
4	30.5	30	-0.5	-1.666666667
5	30.65	30	-0.65	-2.166666667
JARAK 35 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	35.41	35	-0.41	-1.171428571
2	35.51	35	-0.51	-1.457142857
3	35.25	35	-0.25	-0.714285714
4	36.1	35	-1.1	-3.142857143
5	35.4	35	-0.4	-1.142857143

Tabel 4.2 Hasil Pengujian Turun Sensor Ultrasonik

JARAK 35 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	20.53	20	-0.53	-2.65
2	20.56	20	-0.56	-2.8
3	21.1	20	-1.1	-5.5
4	21.05	20	-1.05	-5.25
5	20.65	20	-0.65	-3.25
JARAK 30 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	25.51	25	-0.51	-2.04
2	26.01	25	-1.01	-4.04
3	25.87	25	-0.87	-3.48
4	25.65	25	-0.65	-2.6
5	25.89	25	-0.89	-3.56
JARAK 25 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	30.66	30	-0.66	-2.2
2	31.05	30	-1.05	-3.5
3	30.56	30	-0.56	-1.866666667
4	30.5	30	-0.5	-1.666666667
5	30.65	30	-0.65	-2.166666667
JARAK 20 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	35.41	35	-0.41	-1.171428571
2	35.51	35	-0.51	-1.457142857
3	35.25	35	-0.25	-0.714285714
4	36.1	35	-1.1	-3.142857143
5	35.4	35	-0.4	-1.142857143

Tabel 4.2 Lanjutan

JARAK 15 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	5.51	5	-0.51	-10.2
2	5.38	5	-0.38	-7.6
3	6.12	5	-1.12	-22.4
4	5.64	5	-0.64	-12.8
5	5.83	5	-0.83	-16.6
JARAK 10 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	10.51	10	-0.51	-5.1
2	10.32	10	-0.32	-3.2
3	10.54	10	-0.54	-5.4
4	11.05	10	-1.05	-10.5
5	10.89	10	-0.89	-8.9
JARAK 5 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	15.43	15	-0.43	-2.866666667
2	15.52	15	-0.52	-3.466666667
3	15.62	15	-0.62	-4.133333333
4	15.64	15	-0.64	-4.266666667
5	15.45	15	-0.45	-3

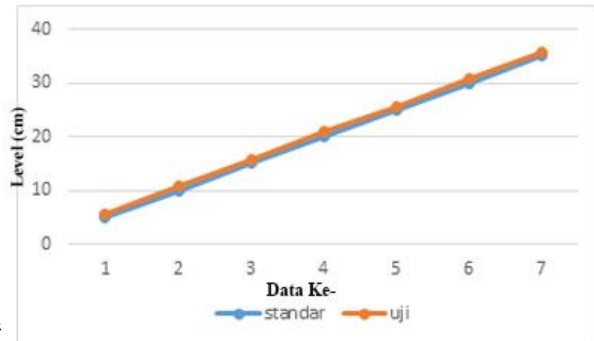
Dari data pembacaan sensor ultrasonic naik dan turun dengan rentang 5 cm sampai dengan 35 cm yang telah didapatkan, maka diperoleh rata-rata pembacaan dari sensor ultrasonic HC-SR04 sebagai berikut:

Tabel 4.3 Rata-rata pembacaan sensor Ultrasonik

JARAK 5 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	5.36	5	-0.36	-7.2
2	5.435	5	-0.435	-8.7
3	5.84	5	-0.84	-16.8
4	5.64	5	-0.64	-12.8
5	5.605	5	-0.605	-12.1
rata rata	5.576			-11.52
JARAK 10 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	10.505	10	-0.505	-5.05
2	10.67	10	-0.67	-6.7
3	10.55	10	-0.55	-5.5
4	10.68	10	-0.68	-6.8
5	10.99	10	-0.99	-9.9
rata rata	10.679			-6.79
JARAK 15 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	15.485	15	-0.485	-3.23
2	15.775	15	-0.775	-5.167
3	15.7	15	-0.7	-4.67
4	15.76	15	-0.76	-5.067
5	15.59	15	-0.59	-3.93
rata rata	15.662			-4.43
JARAK 20 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	20.535	20	-0.535	-2.675
2	20.925	20	-0.925	-4.625
3	21.11	20	-1.11	-5.55
4	20.995	20	-0.995	-4.975
5	20.69	20	-0.69	-3.45
rata rata	20.851			-4.255

Tabel 4.3 Lanjutan

JARAK 25 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	25.535	25	-0.535	-2.14
2	25.855	25	-0.855	-3.42
3	25.46	25	-0.46	-1.84
4	25.78	25	-0.78	-3.12
5	25.89	25	-0.89	-3.56
rata rata	25.704			-2.816
JARAK 30 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	30.585	30	-0.585	-1.95
2	30.82	30	-0.82	-2.733
3	30.53	30	-0.53	-1.7667
4	30.76	30	-0.76	-2.533
5	30.785	30	-0.785	-2.6167
rata rata	30.696			-2.32
JARAK 35 CM				
Data Ke	UJI	STANDAR	KOREKSI	ERROR (%)
1	35.31	35	-0.31	-0.886
2	35.53	35	-0.53	-1.514
3	35.65	35	-0.65	-1.857
4	36.06	35	-1.06	-3.028
5	35.64	35	-0.64	-1.828
rata rata	35.638			-1.822



Gambar 4.3 Grafik Perbandingan Standar dan Uji sensor Ultrasonik

Karakteristik statik adalah karakteristik yang harus diperhatikan apabila alat tersebut digunakan untuk mengukur suatu kondisi yang tidak berubah karena waktu atau hanya berubah secara lambat laun. Untuk itu perlu dilakukan perhitungan untuk mengetahui nilai karakteristik dari sensor ultrasonic. Berikut ini nilai karakteristik statik sensor Ultrasonic:

- Range : 2 cm – 400 cm
- Span : 398 cm
- Resolusi : 0,01 cm
- Sensitivitas : 1.0021 (dapat dilihat pada persamaan 4.1)

$$K(\text{Sensitivitas}) = \frac{\Delta O}{\Delta I}$$

$$= \frac{35,638 - 5,576}{35 - 5} = 1.0021 \dots \dots (4.1)$$

e. *Non – Linieritas*

$$\hat{N} = \frac{[O - KI + a]_{\max}}{O_{\max} - O_{\min}} 100\% \dots \dots \dots (4.2)$$

Dimana :

$$a \text{ (zero bias)} = O_{\min} - KI_{\min} \dots\dots\dots (4.3)$$

$$a = 5.576 - (1.00207)(5)$$

$$a = 0.56565$$

f. Akurasi :

$$A = 1 - \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%, \dots\dots\dots (4.4)$$

$$A = 1 - \left| \frac{35 - 35.638}{35} \right| \times 100\%$$

$$A = 98,178\%$$

dengan Y_n = Pembacaan standar dan X_n = Pembacaan alat

g. Error :

$$e = 1 - A \dots\dots\dots (4.5)$$

$$e = 1 - 0.98178$$

$$e = 0.01822 = 1,822\%$$

h. Histerisis

$$H(I) = O(I)_{I\downarrow} - O(I)_{I\uparrow}, \widehat{H} = H(I)_{\max}$$

sehingga :

$$\% \text{ Maks histeresis} = \frac{\widehat{H}}{O_{\max} - O_{\min}} \times 100\% \dots\dots (4.6)$$

$$\% \text{ Maks histeresis} = \frac{0.26}{30.062} \times 100\%$$

$$\% \text{ Maks histereis} = 0.864\%$$

4.1.4 Pengendalian Motor Servo Operated Valve

Pada perancangan ini, pengendalian motor servo digunakan teknik PWM. Besarnya nilai pulsa on pada teknik PWM akan mempengaruhi gerak daripada motor servo untuk membentuk sudut tertentu. Dari data PWM tersebut akan diketahui berapa bukaan dari valve yang di pasang motor servo. Berikut ini adalah hasil dari pengujian servo :

Tabel 4.4 Hasil Uji Motor Servo Valve

PWM	Bukaan Valve (%)
300	0
310	5.26
320	10.53
330	15.79
340	21.05
350	26.32
360	31.58
370	36.84
380	42.11
390	47.37
400	52.63
410	57.89
420	63.16
430	68.42
440	73.68
450	79.95
460	84.21
470	89.47
480	94.74

4.1.5 Sistem Pengendalian

Pada pengambilan data didapatkan hasil grafik yang menggambarkan respon sistem untuk mencapai set point yang diinginkan . Pengujian dilakukan untuk mengetahui grafik respon actuator dan pembacaan level pada sensor berdasarkan waktu yang telah ditentukan. Sistem pengendalian *level* dikatakan dapat berhasil ketika sistem pengendalian dapat menjaga nilai *level* yang diberikan saat setpoint. Untuk mencapai nilai setpoint yang diberikan dibutuhkan waktu sampai setpoint. Waktu tersebut merupakan nilai yang dibutuhkan untuk mencari nilai respon sistem ketika waktu

sistem pengendalian mencapai setpoint dan mempertahankan setpoint.

4.1.6.1 Hasil respon sistem

Berikut nilai respon sistem pada setpoint tertinggi 26 cm adalah sebagai berikut :

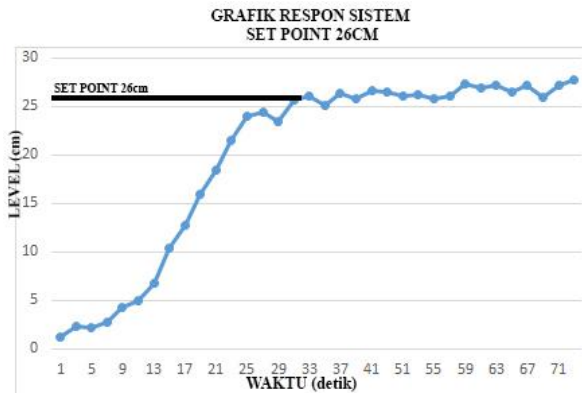
Tabel 4.5 Hasil Uji Set Point 26

Setpoint (cm)	Waktu (s)	Level (cm)
26	1	1.24
	3	2.23
	5	2.13
	7	2.69
	9	4.21
	11	4.89
	13	6.76
	15	10.39
	17	12.74
	19	15.91
	21	18.35
	23	21.53
	25	23.95
	27	24.41
	29	23.38
	31	25.62
	33	26.03
	35	25.09
	37	27.35
	39	25.78
	41	26.62

Tabel 4.5 Lanjutan

	49	27.47
	51	26
	53	26.21

Berdasarkan tabel tersebut didapatkan grafik respon sistem saat sudah mencapai setpoint sebagai berikut ini :

**Gambar 4.4** Grafik respon sistem saat setpoint 26

Berdasarkan grafik tersebut maka didapatkan perhitungan dari respon system tersebut:

a. *Error Steady State*

$$\text{Error} = \frac{26 - 25.62}{26} \times 100\% = 1.055\%$$

b. *Maksimum overshoot*

$$\text{Mp} = \frac{Y_{mp} - Y_{sp}}{Y_{sp}} \times 100\%$$

$$\text{Mp} = \frac{27.47 - 26}{26} \times 100\%$$

$$\text{Mp} = 5.65\%$$

c. *Settling Time*

Waktu yang diperlukan tanggapan sistem untuk mencapai nilai *settling time* menggunakan 5% dari data, sehingga didapatkan nilai sebesar 27 detik

Berikut nilai respon sistem pada setpoint tertinggi 25 cm adalah sebagai berikut :

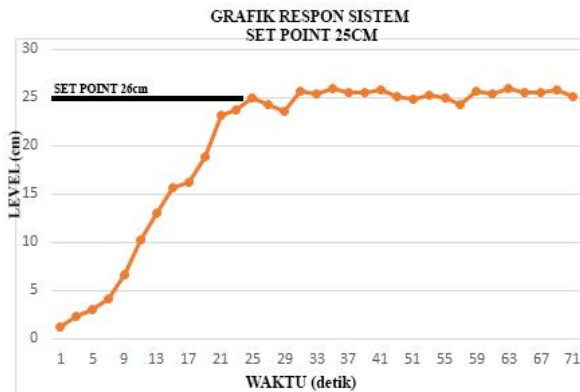
Tabel 4.6 Hasil Uji Set Point 25

Setpoint (cm)	Waktu (s)	Level (cm)
25	1	1.23
	3	2.28
	5	2.97
	7	4.13
	9	6.67
	11	10.21
	13	13.04
	15	15.57
	17	16.15
	19	18.81
	21	23.1
	23	23.7
	25	24.96
	27	24.29
	29	23.58
	31	25.66
	33	25.39
	35	25.93
	37	25.44

Tabel 4.6 Lanjutan

	39	25.44
	41	25.71
	49	25.09
	51	24.78
	53	25.18

Berdasarkan tabel tersebut didapatkan grafik respon sistem saat sudah mencapai setpoint sebagai berikut ini :

**Gambar 4.5** Grafik respon sistem saat setpoint 25

Berdasarkan grafik tersebut maka didapatkan perhitungan dari respon system tersebut

a. *Error Steady State*

$$Error = \frac{25 - 24.41}{24} \times 100\% = 2.45\%$$

b. *Maksimum overshoot*

$$Mp = \frac{Y_{mp} - Y_{sp}}{Y_{sp}} \times 100\%$$

$$Mp = \frac{25.93 - 25}{25} \times 100\%$$

$$Mp = 3.57\%$$

c. *Settling Time*

Waktu yang diperlukan tanggapan sistem untuk mencapai nilai *settling time* menggunakan 5% dari data, sehingga didapatkan nilai sebesar 29 detik

4.2 Pembahasan

Dilakukan pengujian pengendalian level menggunakan MOV. Pengendalian level menggunakan sensor Ultrasonik HC-SR04 yang memiliki range ukur 2cm – 400cm dengan nilai error sebesar 1,822%. MOV pada penelitian kali ini menggunakan motor servo dimana gerakannya berdasarkan dari PWM. Bukaannya dari MOV bergantung pada *set point level*, ketika *level* melebihi dari *set point* maka MOV akan membuka lebih kecil, dan ketika kurang dari *set point* maka bukaan MOV akan bertambah secara otomatis.

Berdasarkan dari pengujian yang telah dilakukan maka didapatkan nilai error *steady state* pada set point 25cm sebesar 2.45% dan pada set point 26cm sebesar 1.055%. Nilai maksimum *overshoot* sebesar 5.65% pada set point 26cm dan 3.57% pada set point 25cm. Waktu yang diperlukan tanggapan sistem untuk mencapai nilai *settling time* menggunakan 5% dari data, sehingga didapatkan nilai sebesar 29 detik pada *set point* 25cm. Pada *set point* 26cm waktu yang diperlukan tanggapan sistem untuk mencapai nilai *settling time* menggunakan 5% dari data, sehingga didapatkan nilai sebesar 27 detik.

Pada pengujian rancang bangun sistem pengendalian level dalam tangki 1 pada sistem hidrostatik skala laboratorium didapatkan nilai tekanan yang sama dengan tangki 2 yaitu sebesar 0,04 bar. Namun nilai laju alirannya kedua tangki berbeda dimana tangki 1 lebih besar daripada tangki 2. Hal ini disebabkan karena ada *orifice plate* yang mempengaruhi kecepatan dari aliran tersebut.

Pada saat alat dijalankan sensor ultrasonik HC-SR04 akan mengukur level air yang kemudian sinyalnya akan diberikan pada mikrokontroler. Namun dalam beberapa pengujian, sensor ultrasonik terkadang tidak membaca sesuai dengan pembacaan meteran ukur pada tabung. Hal ini dikarenakan pada saat pengisian tabung terdapat gelombang air yang dapat mempengaruhi dari pembacaan sensor tersebut.

BAB V

PENUTUP

5.1 Kesimpulan

Adapun kesimpulan yang didapatkan dari tugas akhir kali ini adalah sebagai berikut :

1. Pengendalian level pada tangki menggunakan sensor ultrasonik HC-SR04 untuk membaca ketinggian air pada tangki, sensor telah dikalibrasi dengan nilai error sebesar 1,822% .
2. Dari data respon pengendalian yang didapatkan menunjukkan bahwa pengendalian telah berjalan sesuai *set point* yang diinginkan. Waktu yang dibutuhkan untuk mencapai *set point* 25cm adalah ± 34 detik dan 26cm adalah ± 45 detik.

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan, antara lain:

1. Sebaiknya disediakan sensor cadangan untuk mengantisipasi apabila sensor rusak.
2. Sebaiknya disediakan mikrokontroler cadangan untuk mengantisipasi apabila mikrokontroler rusak.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Abdul M, R. (2017) *Mengenal Motor Servo - Lab Fakultas Ilmu Terapan*. Available at:
<https://fit.labs.telkomuniversity.ac.id/mengenal-motor-servo/>
(Accessed: 24 July 2018).

Adam Thai (2016) *What is Bernoulli's equation? (article) | Khan Academy*. Available at:
<https://www.khanacademy.org/science/physics/fluids/fluid-dynamics/a/what-is-bernoullis-equation> (Accessed: 24 July 2018).

Channels, S. P. W. M. (no date) '– Optional Boot Code Section with Independent Lock Bits – Programming Lock for Software Security and Extended Standby I / O and Packages Microcontroller with 128K Bytes In-System Programmable Flash ATmega1284P'.

Danel, G. (2012) 'OTOMATISASI KERAN DISPENSER BERBASIS MIKROKONTROLER AT89S52 MENGGUNAKAN SENSOR FOTODIODA DAN SENSOR ULTRASONIK PING', 1(1), pp. 60–65.

Haidi, J. (no date) 'Pengaruh Perhitungan Flow Gas Terhadap Perubahan Suhu Gas Alam Dengan Alat Ukur Orifice Meter', pp. 54–61.

Howard Eglowstein (2012) *Introduction to Servo Motors*. Available at: <https://www.sciencebuddies.org/science-fair-projects/references/introduction-to-servo-motors> (Accessed: 24 July 2018).

Ibadurrahman (2017) *Tekanan Hidrostatik - Pengertian, Rumus, Contoh Soal & Pembahasan*. Available at:
<https://www.studiobelajar.com/tekanan-hidrostatik/>
(Accessed: 24 July 2018).

Iwan Nugraha, G. (2014) 'OPTIMALISASI SISTEM PERAWATAN POMPA SENTRIFUGAL DI UNIT UTILITY PT.ABC', 1(1), pp. 77–86.

M. A. Crabtree (2009) "'Industrial Flow Measurement,'" J. Librariansh. Inf. Sci.,' 41.

Martani, M. and Endarko (2014) 'PERANCANGAN DAN PEMBUATAN SENSOR LEVEL UNTUK SISTEM KONTROL PADA PROSES PENGENDAPAN CaCO_3 DALAM AIR DENGAN METODE MEDAN MAGNET', 2(2), pp. 1–5.

Prakoso Galih (2017) *Pengertian Dan Prinsip Kerja Motor Servo - Gagalenyilih*, 23 Juni. Available at: <https://www.gagalenyilih.com/2017/06/pengertian-dan-prinsip-kerja-motor-servo.html> (Accessed: 18 July 2018).

R. H. Perry, D. W. Green, and J. O. M. (1997) 'Chemical Engineers' Handbook Seventh.', 27.

SERVO, M. (2018) 'MKS Servos USA', p. 2018.

Suprianto (2015) *Motor Servo / All Of Life*. Available at: <http://blog.unnes.ac.id/antosupri/motor-servo/> (Accessed: 24 July 2018).

Trisnobudi, A. (2001) *Aplikasi Ultrasonik*. Bandung: Teknik Fisika ITB.

Zimmermann, H. (1999) 'Examination of disturbed pipe flow and its effect on flow measurement using orifice plates, Flow Measurement and Instrumentation.'

LAMPIRAN A
Listing Program pada Code Vision AVR

```
#include <mega1284p.h>
#include <alcd.h>
#include <delay.h>
#include <stdio.h>
#include <stdlib.h>

#define putus 1
#define konek 0

#define nyala 1
#define mati 0

#define dn PINB.0
#define ok PINB.1
#define up PINB.2

#define trb PORTC.2
#define ecb PINC.1

#define tra PORTC.3
#define eca PINC.4

#define vala PORTC.7
#define valb PORTC.6
#define buz PORTC.5
#define inda PORTB.6
#define indb PORTB.7

eeprom float saveFlowKa[100];
eeprom float saveFlowKi[100];
eeprom float saveLevelKa[100];
eeprom float saveLevelKi[100];
```

```
eeeprom int saveSplvl;  
eeeprom int savePeriode;  
eeeprom int saveMov;  
eeeprom int savePmp;
```

```
float flowka,flowki,leveka,levelki;  
int counterka,counterki;  
int cntlvlka,cntlvlki;  
int time,jaraka,jarakaCm;  
float levelka;  
int jaraki,jarakiCm;  
float levelki;  
float vola,voli,presa,presi;  
int msdetik,i,mode,moni;  
unsigned int data,data2;  
int mov,pmp,periode,splvl;  
char  
buff[16],tempflowka[10],tempflowki[10],templevelka[10],te  
mplevelki[10],temppres[10],tempvol[10];  
int mili,detik;
```

```
void bacalevelka();  
void bacalevelki();  
void setpin();  
void setMov(unsigned int persen);  
void setPmp(unsigned int persena);  
void tespmp();  
void tesmov();  
void aktifmov();  
void nonaktifmov();  
void aktifpmp();  
void nonaktifpmp();  
void menu();  
void setting();
```

```
void runsafety();  
void kontrol();
```

```
interrupt [EXT_INT0] void ext_int0_isr(void)  
{  
    counterki++;  
}
```

```
interrupt [EXT_INT1] void ext_int1_isr(void)  
{  
    counterka++;  
}
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)  
{  
    TCNT0=0x64;  
    mili++;  
    if(mili>10){  
        flowka = (float) counterka*0.1/4.8;  
        flowki = (float) counterki*0.1/4.8;  
  
        bacalevelka();  
        bacalevelki();
```

```
        vola    = (float) 3.14*0.01*levelka*10;  
        presa   = (float) leveka/100;  
        voli    = (float) 3.14*0.01*levelki*10;  
        presi   = (float) levelki/100;
```

```
    kontrol();
```

```
    mili=0;  
    msdetik++;  
    if(msdetik>=10){
```

```

    detik++;
    i++;
    if(detik>=periode){
        flowka=saveFlowKa[i];
        flowki=saveFlowKi[i];
        levelka=saveLevelKa[i];
        levelki=saveLevelKi[i];
        detik=0;

        printf("upmoni.php?lvla="+String(levela)+"&vola="+String(vola)+"&floa="+String(flowa)+"&prea="+String(presa)+"&lvlb="+String(levelb)+"&volb="+String(volb)+"&flob="+String(flowb)+"&preb="+String(presb)+"&mov="+String(mov)+"&mtr="+String(mtr));
    }
}

    if(!up){moni++; delay_ms(50);}
    if(!dn){moni--; delay_ms(50);}
    if(!ok){
        delay_ms(50);
        lcd_clear();
        lcd_putsf("Saving EEPROM...");
        for(i=0;i<20;i++){
            lcd_gotoxy(i,1);
            lcd_putchar(0xff);
            delay_ms(100);
        }
        lcd_clear();
        for(;;){
            lcd_gotoxy(0,0);
            lcd_putsf("Save Ok");
            lcd_gotoxy(0,1);
            lcd_putsf("Tekan tombol hitam");
        }
    }

```



```

}
if(moni<1) {moni=4;}
if(moni>4) {moni=1;}

```

```

if(moni==1){
    lcd_gotoxy(8,0); lcd_putsf("Kiri ");
    lcd_gotoxy(0,1); lcd_putsf("Mov :    % ");
    lcd_gotoxy(6,1); sprintf(buff,"%3d",mov); lcd_puts(buff);
    lcd_gotoxy(0,2); lcd_putsf("Flw :    L/s");
    lcd_gotoxy(6,2); ftoa(flowki,2,tempflowki);
    lcd_puts(tempflowki);
    lcd_gotoxy(0,3); lcd_putsf("Lvl :    cm ");
    lcd_gotoxy(6,3); ftoa(levelki,2,templevelki);
    lcd_puts(templevelki);
}

```

```

else if(moni==2){
    lcd_gotoxy(8,0); lcd_putsf("Kiri ");
    lcd_gotoxy(0,1); lcd_putsf("Prs :    kpa");
    lcd_gotoxy(6,1); ftoa(presi,2,temppres);
    lcd_puts(temppres);
    lcd_gotoxy(0,2); lcd_putsf("Vol :    L ");
    lcd_gotoxy(6,2); ftoa(voli,2,tempvol); lcd_puts(tempvol);
    lcd_gotoxy(0,3); lcd_putsf("    ");
}

```

```

else if(moni==3){
    lcd_gotoxy(8,0); lcd_putsf("Kanan");
    lcd_gotoxy(0,1); lcd_putsf("Pmp :    % ");
    lcd_gotoxy(6,1); sprintf(buff,"%3d",pmp); lcd_puts(buff);
    lcd_gotoxy(0,2); lcd_putsf("Flw :    L/s");
    lcd_gotoxy(6,2); ftoa(flowka,2,tempflowka);
    lcd_puts(tempflowka);
    lcd_gotoxy(0,3); lcd_putsf("Lvl :    cm ");
}

```

```

    lcd_gotoxy(6,3); ftoa(levelka,2,templevelka);
    lcd_puts(templevelka);
}

```

```

else if(moni==4){
    lcd_gotoxy(8,0); lcd_putsf("Kanan");
    lcd_gotoxy(0,1); lcd_putsf("Prs :    kpa");
    lcd_gotoxy(6,1); ftoa(presa,2,temppres);
    lcd_puts(temppres);
    lcd_gotoxy(0,2); lcd_putsf("Vol :    L ");
    lcd_gotoxy(6,2); ftoa(vola,2,tempvol); lcd_puts(tempvol);
    lcd_gotoxy(0,3); lcd_putsf("          ");
}

```

```

}

```

```

}

```

```

void setpin(){
    DDRB.0=0; PORTB.0=1; // dn
    DDRB.1=0; PORTB.1=1; // ok
    DDRB.2=0; PORTB.1=1; // up

    DDRD.2=0; PORTD.2=1; // Interrupt Flowkanan
    DDRD.3=0; PORTD.3=1; // Interrupt Flowkiri

    DDRC.2=1; PORTC.2=0; // Trigger level kiri
    DDRC.1=0; PORTC.1=1; // Echo level kiri

    DDRC.3=1; PORTC.3=0; // Trigger level kanan
    DDRC.4=0; PORTC.4=1; // Echo level kanan

    DDRC.7=1; PORTC.7=putus; // Valve kanan
    DDRC.6=1; PORTC.6=putus; // Valve kiri
}

```

```
DDRC.5=1; PORTC.5=putus; // Bel
```

```
DDRB.7=1; PORTB.7=mati; // indikator kanan
```

```
DDRB.6=1; PORTB.6=mati; // indikator kiri
```

```
}
```

```
void bacalevelka(){
```

```
    tra=1;
```

```
    delay_us(10);
```

```
    tra=0;
```

```
    for(;;){
```

```
        if(eca==1){
```

```
            cntlvlka=0;
```

```
            break;
```

```
        }
```

```
    }
```

```
    for(;;){
```

```
        cntlvlka++;
```

```
        if(eca==0){
```

```
            break;
```

```
        }
```

```
    }
```

```
    time = (float) 0.0000000625*cntlvlka;
```

```
    jaraka = (float) (time*343)/2;
```

```
    jarakaCm = (float) (cntlvlka*31.133+26.98)/1000;
```

```
    levelka = (float) 50-jarakaCm;
```

```
    if(levelka<0){ levelka=0;}
```

```
}
```

```
void bacalevelki(){
```

```
    trb=1;
```

```

delay_us(10);
trb=0;

for(;;){
    if(ecb==1){
        cntlvlki=0;
        break;
    }
}

for(;;){
    cntlvlki++;
    if(ecb==0){
        break;
    }
}

time = (float) 0.0000000625*cntlvlki;
jaraki = (float) (time*343)/2;
jarakiCm = (float) (cntlvlki*31.133+26.98)/1000;
levelki = (float) 50-jarakiCm;
if(levelki<0){ levelki=0;}
}

void setMov(unsigned int persen){
    data = persen*1.9+300;
    OCR1AH = (unsigned char)(((data & (unsigned
int)0xFF00))>>8);
    OCR1AL = (unsigned char) (data & (unsigned
int)0x00FF);
}

void setPmp(unsigned int persena){
    data2 = persena*2.5+250;

```

```
OCR1BH = (unsigned char)(((data2 & (unsigned
int)0xFF00))>>8);
```

```
OCR1BL = (unsigned char) (data2 & (unsigned
int)0x00FF);
}
```

```
void aktifmov(){
    DDRD.5=1;
    TCCR1A=0x82;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x03;
    ICR1L=0x02;
}
```

```
void nonaktifmov(){
    DDRD.5=0;
    TCCR1A=0x02;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x03;
    ICR1L=0x02;
}
```

```
void aktifpmp(){
    DDRD.4=1;
    TCCR1A=0x22;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x13;
    ICR1L=0x87;
}
```

```

void nonaktifpmp(){
    PORTD.4=1;
    TCCR1A=0x00;
    TCCR1B=0x00;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x00;
    ICR1L=0x00;
}

```

```

void tesvalve(){
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("Tes Valve");
    for(;;){
        if(!up){
            lcd_gotoxy(0,1);
            lcd_putsf("Valve kiri ");
            delay_ms(300);
            vala=konek;
            valb=putus;
        }

        if(!dn){
            lcd_gotoxy(0,1);
            lcd_putsf("Valve kanan");
            delay_ms(300);
            vala=konek;
            valb=putus;
        }
    }
}

```

```

    lcd_clear();

```

```

}

void tesmov(){
    i=35;
    aktifmov();
    for(;;){
        if(!up){ i++; delay_ms(100);}
        if(!dn){ i--; delay_ms(100);}
        if(!ok){ delay_ms(200); lcd_clear(); break; }
        if(i>100) {i=10; delay_ms(10);}
        if(i<10) {i=100; delay_ms(10);}
        data = i*1.9+300;
        //OCR1AH = (unsigned char)(((data & (unsigned
int)0xFF00))>>8);
        //OCR1AL = (unsigned char) (data & (unsigned
int)0x00FF);
        setMov(i);
        sprintf(buff,"MOV : %4d %%",i);
        lcd_gotoxy(0,0);
        lcd_puts(buff);
        sprintf(buff,"PWM : %4d ",data);
        lcd_gotoxy(0,1);
        lcd_puts(buff);
    }
    nonaktifmov();
}
}

```

```

void tespmp(){
    i=35;
    aktifpmp();
    for(;;){
        if(!up){ i++; delay_ms(100);}
        if(!dn){ i--; delay_ms(100);}
        if(!ok){ delay_ms(200); lcd_clear(); break; }
    }
}

```

```

        if(i>100) {i=10; delay_ms(10);}
        if(i<10) {i=100; delay_ms(10);}
        data2 = i*2.5+250;
        //OCR1BH = (unsigned char)(((data & (unsigned
int)0xFF00))>>8);
        //OCR1BL = (unsigned char) (data & (unsigned
int)0x00FF);
        setPmp(i);
        sprintf(buff,"MTR : %4d %%",i);
        lcd_gotoxy(0,0);
        lcd_puts(buff);
        sprintf(buff,"PWM : %4d ",data2);
        lcd_gotoxy(0,1);
        lcd_puts(buff);
    }
    nonaktifpmp();

}

void menu(){
    awal:
    lcd_gotoxy(0,0); lcd_putsf("Main menu");
    lcd_gotoxy(2,1); lcd_putsf("Data EEPROM");
    lcd_gotoxy(2,2); lcd_putsf("Running");

    menu1:
    lcd_gotoxy(0,1); lcd_putchar(0x7E);
    if(!dn){
        delay_ms(250); lcd_clear(); goto awal2;
    }
    if(!ok){
        delay_ms(250);
        lcd_gotoxy(0,1); lcd_putchar(0x20);
        goto menu2;
    }
}

```



```

else goto menu1;

menu2:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!dn){
    delay_ms(250); lcd_clear(); goto awal3;
}
if(!ok){
    delay_ms(250);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu1;
}
else goto menu2;

awal2:
i=0;
lcd_gotoxy(0,0); lcd_putsf("EEPROM flow");
lcd_gotoxy(2,1); sprintf(buff,"Ki [%2d] :    L/s",i);
lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"Ka [%2d] :    L/s",i);
lcd_puts(buff);
lcd_gotoxy(2,3); lcd_putsf("level");
lcd_gotoxy(11,1); ftoa(saveFlowKa[i],2,tempflowka);
lcd_puts(tempflowka);
lcd_gotoxy(11,2); ftoa(saveFlowKi[i],2,tempflowki);
lcd_puts(tempflowki);

lihat1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){i++; delay_ms(200);}
if(!dn){i--; delay_ms(200);}
if(i>99) {i=0;}
if(i<0) {i=99;}
lcd_gotoxy(2,1); sprintf(buff,"Ki [%2d] :    L/s",i);
lcd_puts(buff);

```

```

    lcd_gotoxy(2,2);  sprintf(buff,"Ka [%2d] :    L/s",i);
    lcd_puts(buff);
    lcd_gotoxy(11,1);  ftoa(saveFlowKa[i],2,tempflowka);
    lcd_puts(tempflowka);
    lcd_gotoxy(11,2);  ftoa(saveFlowKi[i],2,tempflowki);
    lcd_puts(tempflowki);

```

```

    if(!ok){
        delay_ms(200);
        lcd_gotoxy(0,1); lcd_putchar(0x20);
        goto lihat2;}
    else goto lihat1;

```

```

    lihat2:
    lcd_gotoxy(0,3); lcd_putchar(0x7E);
    if(!dn){
        delay_ms(200);
        lcd_clear();
        lcd_gotoxy(0,3); lcd_putchar(0x20);
        goto levelawal2;}

```

```

    if(!ok){
        delay_ms(200);
        lcd_gotoxy(0,3); lcd_putchar(0x20);
        goto lihat1;}
    else goto lihat2;

```

```

    levelawal2:
    i=0;
    lcd_gotoxy(0,0);  lcd_putsf("EEPROM Level");
    lcd_gotoxy(2,1);  sprintf(buff,"Li [%2d] :    cm",i);
    lcd_puts(buff);
    lcd_gotoxy(2,2);  sprintf(buff,"La [%2d] :    cm",i);
    lcd_puts(buff);
    lcd_gotoxy(2,3);  lcd_putsf("back");

```

```
    lcd_gotoxy(11,1); ftoa(saveLevelKa[i],2,templevelka);  
    lcd_puts(templevelka);  
    lcd_gotoxy(11,2); ftoa(saveLevelKi[i],2,templevelki);  
    lcd_puts(templevelki);
```

```
    levellihat1:  
    lcd_gotoxy(0,1); lcd_putchar(0x7E);  
    if(!up){i++; delay_ms(200);}   
    if(!dn){i--; delay_ms(200);}   
    if(i>99) {i=0;}   
    if(i<0) {i=99;}   
    lcd_gotoxy(2,1); sprintf(buff,"Li [%2d] :    cm",i);  
    lcd_puts(buff);  
    lcd_gotoxy(2,2); sprintf(buff,"La [%2d] :    cm",i);  
    lcd_puts(buff);  
    lcd_gotoxy(11,1); ftoa(saveLevelKa[i],2,templevelka);  
    lcd_puts(templevelka);  
    lcd_gotoxy(11,2); ftoa(saveLevelKi[i],2,templevelki);  
    lcd_puts(templevelki);
```

```
    if(!ok){  
        delay_ms(200);  
        lcd_gotoxy(0,1); lcd_putchar(0x20);  
        goto levellihat2;}  
    else goto levellihat1;
```

```
    levellihat2:  
    lcd_gotoxy(0,3); lcd_putchar(0x7E);  
    if(!dn){  
        delay_ms(200);  
        lcd_clear();  
        lcd_gotoxy(0,3); lcd_putchar(0x20);  
        goto awal;}  
    else goto awal;
```

```
    if(!ok){
```

```
    delay_ms(200);  
    lcd_gotoxy(0,3); lcd_putchar(0x20);  
    goto levellihat1;}  
else goto levellihat2;
```

```
awal3:  
lcd_gotoxy(0,0); lcd_putsf("Set mode :");  
mode=0;
```

```
menumode:  
if(!dn){mode=0;}  
if(!up){mode=1;}  
if(mode==0){lcd_gotoxy(11,0); lcd_putsf("Stabil ");}  
if(mode==1){lcd_gotoxy(11,0); lcd_putsf("Safety ");}
```

```
if(!ok){  
    delay_ms(200); goto exit1;  
}  
else goto menumode;
```

```
exit1:  
lcd_gotoxy(0,1); lcd_putsf("Oke...");  
delay_ms(500);  
lcd_clear();  
}
```

```
void setting(){  
    lcd_gotoxy(0,0); lcd_putsf("Set Mov & Pmp");  
    lcd_gotoxy(2,1); sprintf(buff,"Mov : %3d  %%",mov);  
    lcd_puts(buff);  
    lcd_gotoxy(2,2); sprintf(buff,"Pmp : %3d  %%",pmp);  
    lcd_puts(buff);  
    lcd_gotoxy(2,3); lcd_putsf("Oke");
```

```
menu1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){mov++; delay_ms(150);}
if(!dn){mov--; delay_ms(150);}
if(mov>100){mov=0;}
if(mov<30){mov=100;}
lcd_gotoxy(2,1); sprintf(buff,"Mov : %3d  %%",mov);
lcd_puts(buff);
saveMov=mov;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,1); lcd_putchar(0x20);
    goto menu2;
}
goto menu1;
```

```
menu2:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!up){pmp++; delay_ms(150);}
if(!dn){pmp--; delay_ms(150);}
if(pmp>100) {pmp=0;}
if(pmp<30) {pmp=100;}
lcd_gotoxy(2,2); sprintf(buff,"Pmp : %3d  %%",pmp);
lcd_puts(buff);
savePmp=pmp;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu3;
}
goto menu2;
```

```
menu3:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
```

```

if(!dn){
    lcd_gotoxy(6,3); lcd_putsf("Saving");
    delay_ms(200); lcd_clear(); goto exit1;
}
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto menu1;
}
goto menu3;

exit1:
    lcd_gotoxy(0,0); lcd_putsf("Set Time & Level");
    lcd_gotoxy(2,1); sprintf(buff,"Sve : %3d dt",periode);
    lcd_puts(buff);
    lcd_gotoxy(2,2); sprintf(buff,"Spl : %3d cm",splvl);
    lcd_puts(buff);
    lcd_gotoxy(2,3); lcd_putsf("Oke");

menu4:
    lcd_gotoxy(0,1); lcd_putchar(0x7E);
    if(!up){periode++; delay_ms(150);}
    if(!dn){periode--; delay_ms(150);}
    if(periode>360){periode=0;}
    if(periode<0) {periode=360;}
    lcd_gotoxy(2,1); sprintf(buff,"Sve : %3d dt",periode);
    lcd_puts(buff);
    savePeriode=periode;
    if(!ok){
        delay_ms(200);
        lcd_gotoxy(0,1); lcd_putchar(0x20);
        goto menu5;
    }
    goto menu4;

```

```

menu5:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!up){splvl++; delay_ms(150);}
if(!dn){splvl--; delay_ms(150);}
if(splvl>45) {splvl=0;}
if(splvl<0) {splvl=45;}
lcd_gotoxy(2,2); sprintf(buff,"Spl : %3d cm",splvl);
lcd_puts(buff);
saveSplvl=splvl;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu6;
}
goto menu5;

```

```

menu6:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
if(!dn){
    lcd_gotoxy(6,3); lcd_putsf("Saving");
    delay_ms(200); lcd_clear(); goto exit2;
}
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto menu4;
}
goto menu6;

```

```

exit2:
lcd_clear();
}

```

```

void runsafety(){

```

```

lcd_gotoxy(0,0);
lcd_putsf("Preparing MOV ");
aktifmov();
setMov(40);
for(i=0;i<20;i++){
    lcd_gotoxy(i,1);
    lcd_putchar(0xff);
    delay_ms(100);
}
nonaktifmov();
lcd_clear();

lcd_gotoxy(0,0);
lcd_putsf("Preparing PMP ");
aktifpmp();
setPmp(40);
for(i=0;i<20;i++){
    lcd_gotoxy(i,1);
    lcd_putchar(0xff);
    delay_ms(100);
}
nonaktifpmp();
lcd_clear();
lcd_gotoxy(0,0) ; lcd_putsf("Tes Safety");
for(;;){
    bacalevela();
    bacalevelb();
    lcd_gotoxy(0,1); lcd_putsf("Lki :    cm ");
    lcd_gotoxy(6,1); ftoa(levelki,2,templevelki);
    lcd_puts(templevelki);
}
    lcd_gotoxy(0,2); lcd_putsf("Lka :    cm ");
    lcd_gotoxy(6,2); ftoa(levelka,2,templevelka);
    lcd_puts(templevelka);
}

```



```

// safety kanan
if(levelb >= 34){
    buz=konek; delay_ms(500); buz=putus; delay_ms(500);
    valb=konek; delay_ms(500);
    aktifpmp();
    setPmp(10);delay_ms(500);
    nonaktifpmp();
}

if(levelb>13){
    indb=mati;
}

if(levelb<10){
    indb=nyala;
    valb=putus; delay_ms(500);
    aktifpmp();
    setPmp(90);delay_ms(500);
    nonaktifpmp();
}

// safety kiri
if(levela >= 34){
    buz=konek; delay_ms(500);    buz=putus; delay_ms(500);
    vala=konek; delay_ms(500);
    aktifmov();
    setMov(10);delay_ms(500);
    nonaktifmov();
}

if(levela>13){
    inda=mati;
}

```

```

    if(levela<10){
        inda=nyala;
        vala=putus; delay_ms(500);
        aktifmov();
        setMov(90);delay_ms(500);
        nonaktifmov();
    }

    delay_ms(200);
}
}

void kontrol(){
    if(levelki >= 34 || levelka>=34){
        buz=konek; delay_ms(500); buz=putus;
    }

    if(levelki < 14){
        indb=nyala;
    }

    if(levelki >= 20){
        indb=mati;
    }

    if(levelka < 14){
        inda=nyala;
    }

    if(levelka >= 20){
        inda=mati;
    }

    if(levelki>=splvl+1){
        valb=konek;
    }

```

```
}
```

```
if(levelki<splvl-1){  
    valb=putus;  
}
```

```
if(levelka>=splvl+1){  
    vala=konek;  
}
```

```
if(levelka<splvl){  
    vala=putus;  
}
```

```
// Kontrol kanan  
if(levelki > splvl+5){  
    pmp=savePmp;  
    pmp=10;  
    aktifpmp();  
    setPmp(10);delay_ms(500);  
    nonaktifpmp();  
}
```

```
if(levelki < splvl+5 && levelki >= splvl+2){  
    pmp = savePmp;  
    pmp=30;  
    aktifpmp();  
    setPmp(30);delay_ms(500);  
    nonaktifpmp();  
}
```

```
if(levelki < splvl+2 && levelki >= splvl-2){  
    pmp = savePmp;  
    aktifpmp();
```

```
setPmp(pmp);delay_ms(500);  
nonaktifpmp();  
}
```

```
if(levelki < splvl-2 && levelki >= splvl-5){  
pmp=70;  
aktifpmp();  
setPmp(70);delay_ms(500);  
nonaktifpmp();  
}
```

```
if(levelki <= splvl-5){  
pmp=90;  
aktifpmp();  
setPmp(90);  
delay_ms(500);  
nonaktifpmp();  
}
```

// Kontrol Kiri

```
if(levelka > splvl+5){  
mov=10;  
aktifmov();  
setMov(10); delay_ms(500);  
nonaktifmov();  
}
```

```
if(levelka < splvl+5 && levelka >= splvl+2){  
mov=30;  
aktifmov();  
setMov(30);delay_ms(500);  
nonaktifmov();  
}
```

```
if(levelka < splvl+2 && levelka >= splvl-2){  
    mov=saveMov;  
    aktifmov();  
    setMov(mov);delay_ms(500);  
    nonaktifmov();  
}
```

```
if(levelka < splvl-2 && levelka >= splvl-5){  
    mov=70;  
    aktifmov();  
    setMov(70);delay_ms(500);  
    nonaktifmov();  
}
```

```
if(levelka <= splvl-5){  
    mov=90;  
    aktifmov();  
    setMov(90); delay_ms(500);  
    nonaktifmov();  
}  
}
```

```
void main(void)  
{
```

```
    setpin();
```

```
    TCCR0A=0x00;  
    TCCR0B=0x05;  
    TCNT0=0x64;  
    OCR0A=0x00;  
    OCR0B=0x00;
```

```
    TCCR1A=0xA2;
```

```
TCCR1B=0x1B;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x13;  
ICR1L=0x87;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
EICRA=0x0B;  
EIMSK=0x03;  
EIFR=0x03;  
PCICR=0x00;
```

```
TIMSK0=0x01;  
TIMSK1=0x00;  
TIMSK2=0x00;
```

```
UCSR0A=0x00;  
UCSR0B=0x08;  
UCSR0C=0x06;  
UBRR0H=0x00;  
UBRR0L=0x67;
```

```
pmp = savePmp;  
mov = saveMov;
```

```
lcd_init(16);  
lcd_putsf("Loading EEPROM... ");  
for(i=0;i<20;i++){  
    lcd_gotoxy(i,1);  
    lcd_putchar(0xff);  
    delay_ms(100);  
}
```

```
lcd_clear();  
if(!dn){delay_ms(350); tesmov();}  
if(!ok){delay_ms(350); tesvalve();}  
if(!up){delay_ms(350); tespmp();}
```

```
menu();
```

```
if(mode==0){setting();}  
if(mode==1){runsafety();}
```

```
lcd_gotoxy(0,0);  
lcd_putsf("Preparing MOV ");  
aktifmov();  
setMov(mov);  
for(i=0;i<20;i++){  
    lcd_gotoxy(i,1);  
    lcd_putchar(0xff);  
    delay_ms(100);  
}  
nonaktifmov();  
lcd_clear();
```

```
lcd_gotoxy(0,0);  
lcd_putsf("Preparing PMP ");  
aktifpmp();  
setPmp(pmp);  
for(i=0;i<20;i++){  
    lcd_gotoxy(i,1);  
    lcd_putchar(0xff);  
    delay_ms(100);  
}  
nonaktifpmp();  
lcd_clear();
```

```
moni=1;
```

```
lcd_gotoxy(0,0); lcd_putsf("Plant : ");  
printf("uphistory.php");  
delay_ms(200);  
msdetik=0; mili=0;  
#asm("sei")
```

```
while (1)  
{  
  
}  
}
```


LAMPIRAN B

Data Sheet ATMEGA1284P

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 128K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 4K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 16K Bytes Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel, 10-bit ADC
 - Differential mode with selectable gain at 1x, 10x or 20x
 - Byte-oriented Two-wire Serial Interface
 - Two Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 1.8 - 5.5V for ATmega1284P
- Speed Grades
 - 0 - 4 MHz @ 1.8 - 5.5V
 - 0 - 10 MHz @ 2.7 - 5.5V
 - 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active: 0.4 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.7 µA (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 128K Bytes
In-System
Programmable
Flash

ATmega1284P

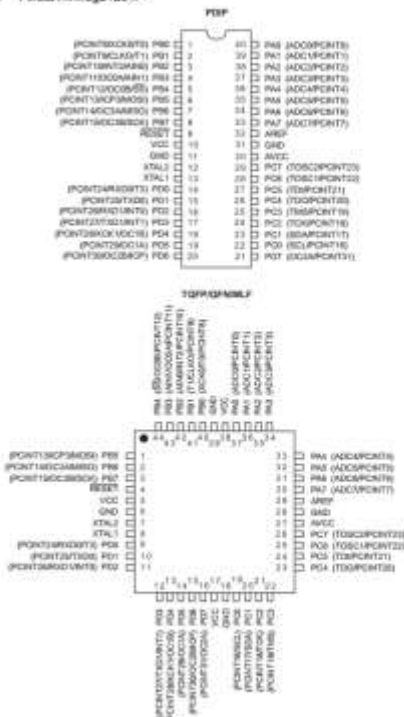
Preliminary

80000-000-11/00



1. Pin Configurations

Figure 1-1. Pinout ATmega1284P



Note: The large center pad underneath the CPMMLF package should be soldered to ground on the board to ensure good mechanical stability.

LAMPIRAN C

Data Sheet Ultrasonic HC-SR04



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time*velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

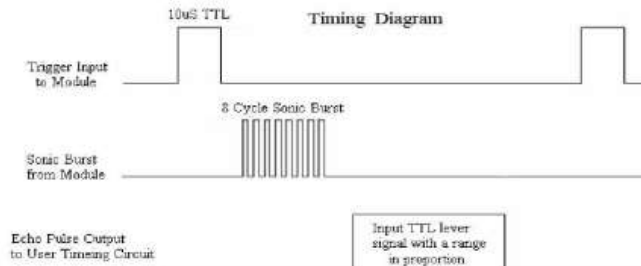
Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{s} / 58 = \text{centimeters}$ or $\mu\text{s} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.ElecFreaks.com



LAMPIRAN D

Data Sheet Servo MKS HV777

700018

MKS Servo USA :: High Voltage Coreless Servo :: HV777 Aluminum upper Casing (0.1 sec/60Å° - 38.00 kg.cm (520 oz/in) @8.2V)



MKS Servos USA

CALL US: 832.287.2271

Home :: High Voltage Coreless Servos :: HV777 Aluminum upper Casing (0.1 sec/60Å° - 38.00 kg.cm (520 oz/in) @8.2V)

HV777 Aluminum upper Casing (0.1 sec/60Å° - 38.00 kg.cm (520 oz/in) @8.2V)



HV777

Details

SKU	RC-MKS-hv777
Dead Band	0.0000 ms (Default)
Control System	+Pulse Width Control
Working Frequency	1500us / 333Hz
(Pin) Required Pulse	3.5 ~8.4 Volt Peak to Peak Square Wave
Operating Voltage	6.0 ~7.4 V DC Voltage
Operating Temperature Range	-10 to + 60 Degree C
Operating Speed (0V)	0.14 sec/60Å° degrees at no load
Operating Speed (7.4V)	0.11 sec/60Å° degrees at no load
Operating Speed (8.4V)	0.10 sec/60Å° degrees at no load
Stall Torque (0V)	38.00 kg/cm
Stall Torque (7.4V)	36.5 kg/cm
Stall Torque (8.4V)	38.00 kg/cm
Motor Type	Coreless Motor
Potentiometer Drive	Indirect Drive
Driver Type	FET
Bearing Type	Dual Ball Bearings
Gear Type	Titanium Gears
Programmable	NO
Connector Wire Length	15.0 cm (5.9 in)
Dimensions	40x20x45 mm (1.57x0.87x1.77 in)
Weight	81.4 g

Our price: **\$106.99**

Options

Quantity: